REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	21 March 2000	Symposium Paper 21-23 March 2000

4. TITLE AND SUBTITLE
System Performance Modeling in C4ISR/Weapon System Design and Development

5. FUNDING NUMBERS

6. AUTHOR(S)
Stephan J. Froggett and Simon Liu

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

ComGlobal Systems Inc
1111 Jefferson Davis Highway
Suite 510
Arlington VA 22202

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Naval Surface Warfare Center
Dahlgren Division
17320 Dahlgren Road
Code N10
Dahlgren VA 22448-5100

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

N/A

11. SUPPLEMENTARY NOTES
Prepared for the Engineering the Total Ship (ETS) 2000 Symposium held in Gaithersburg, Md. at the National Institute of Standards & Technology and sponsored by the Naval Surface Warfare Center & the American Society of Naval Engineers

12a. DISTRIBUTION / AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE
Approved for public release: Distribution is unlimited	A

13. ABSTRACT ***(Maximum 200 Words)***
This paper describes an effective process for development of engineering models and discrete event simulations as part of the system engineering effort supporting Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) and weapon system development. Application of modeling and simulation techniques throughout the system life cycle has been directed as an element of Defense acquisition reform, and has been demonstrated to be effective in reducing cost, risk, and improving system performance. Development of an executable model of the proposed system, which encompasses the functional architecture, process models, rules, and a data representation, allows the architect to ensure the design concept meets functional requirements. When this is carried a step further by developing a simulation of the architecture, or architectural components, it becomes possible to assess performance capabilities. A virtual model of the system can be executed to predict these characteristics and validate its likely fulfillment of operational requirements. This paper provides a step-by-step discussion of a process for developing system performance models and simulations, concluding with a synopsis of key areas for program manager attention.

14. SUBJECT TERMS	15. NUMBER OF PAGES
modeling; simulation; performance assessment; system engineering	12
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

AQ U00-07-1833

Steve Froggett and Simon Liu, Com Global Systems Inc.

System Performance Modeling in C4ISR/Weapon System Design and Development

ABSTRACT

This paper describes an effective process for development of engineering models and discrete event simulations as part of the system engineering effort supporting Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) and weapon system development. Application of modeling and simulation techniques throughout the system life cycle has been directed as an element of Defense acquisition reform, and has been demonstrated to be effective in reducing cost, risk, and improving system performance. Development of an executable model of the proposed system, which encompasses the functional architecture, process models, rules, and a data representation, allows the architect to ensure the design concept meets functional requirements. When this is carried a step further by developing a simulation of the architecture, or architectural components, it becomes possible to assess performance capabilities. A virtual model of the system can be executed to predict these characteristics and validate its likely fulfillment of operational requirements. This paper provides a step-by-step discussion of a process for developing system performance models and simulations, concluding with a synopsis of key areas for program manager attention.

INTRODUCTION

Models are abstract representations of systems, providing a logical description of how the system works. Models may be used to provide insight into system structure, operation, and its interaction with the environment. When the model is embodied in software and is executable on a computer, system behavior and performance under various conditions and alternate system concepts can be examined before significant resources are expended in system development. Modeling allows system designers to experiment with the system processes without actually having the system, or having to create complex real world environments to interact with the system.

MODELING IN SYSTEM ENGINEERING

DoD Policy

Appropriate use of modeling and simulation techniques is mandated as a major element of DoD acquisition policy. As stated in DoD Directive 5000.1, modeling and simulation shall be used to reduce the time, resources, and risks associated with the acquisition process while improving the quality of the system being acquired. The Simulation Test and Evaluation Process (STEP), a major DoD initiative, focuses on use of modeling and simulation in conjunction with test and evaluation throughout the system life cycle. DoD STEP Guidelines note that "Credible representations of the system and simulations can provide early and continuous insight and projections and predictions about system performance; risk and risk mitigation; operational effectiveness, survivability, and suitability; and to support others in the acquisition, requirements, cost analysis, training, and user communities." (Coyle and Sanders, 1997) Figure 1 illustrates the concept that

knowledge gained from the application of the STEP processes of simulation, testing, analysis and evaluation permits an informed assessment of system ability to meet requirements.

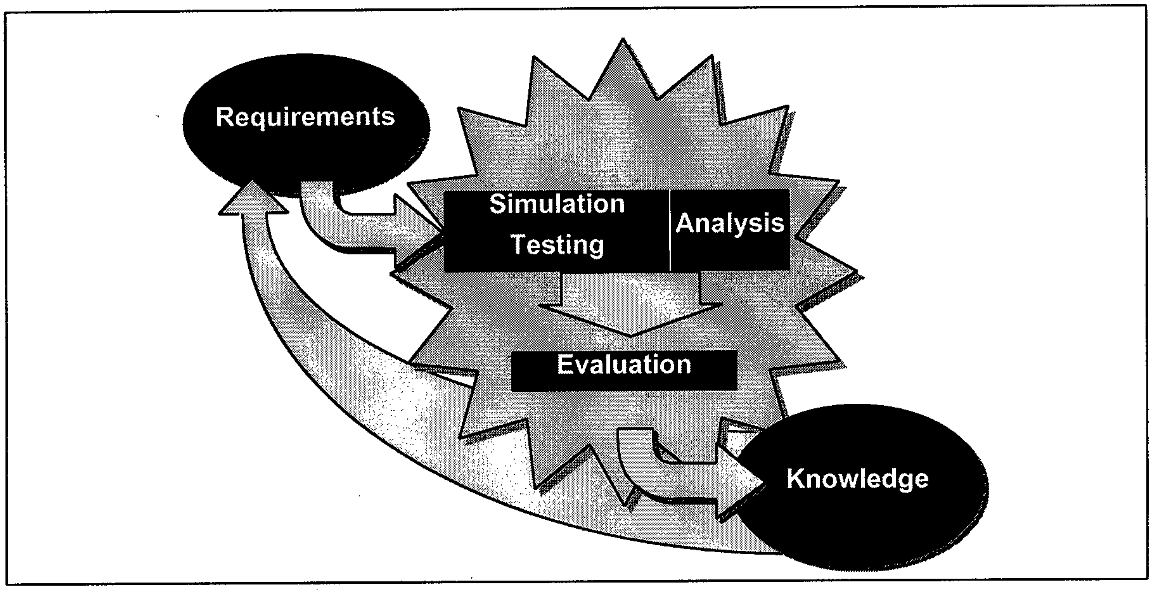

Figure 1. STEP Evaluation Process

System Development and Architecture Modeling

The SPAWAR C4ISR Architecture Framework defines a means of expressing an architecture in multiple ways, each providing a different insight into the system. The SPAWAR approach begins with an analysis of the operational mission and supporting tasks, which are decomposed into the functions which must be performed to achieve the tasks and accomplish the mission. The operational concept drives the organization or operational architecture (system nodes and relationships), and impacts the physical (system) architecture, which provides the resources to accomplish the required functions. The functional architecture identifies the functions that are to be performed, as well as their sequence, control mechanisms, inputs, and outputs. The functional architecture may be expressed in the form of several models, including:

- Process models, such as functional or data flow diagrams which describe how the system is to perform in terms of the functions that have been identified in the modeling process;
- Data models, describing data structures and relationships among the data entities that have been identified in the process models;
- Rule models which describe high level behavior for transformation of inputs or controls to outputs for functions identified in the process models;
- Dynamics models describing the conditions that cause the system to transition from one state to another.

Each of these views defines a different aspect of the system, and contributes to an understanding of how the system should perform. However, these are all static views, and do not provide insight into dynamic interactions among system elements. When the processes, rules, data flows, and dynamics of a system have been identified they constitute a system functional architecture. Creation of an executable performance model involves marrying the functional architecture with at least some aspects of a physical architecture and defined organizational relationships, with

the whole thing driven by the concept of operations. When this is carried one step further, by adding performance values and probabilistic responses to build a dynamic simulation, a significantly greater understanding of likely system performance under operational conditions may be gained. This is an essential element in the engineering of complex systems.

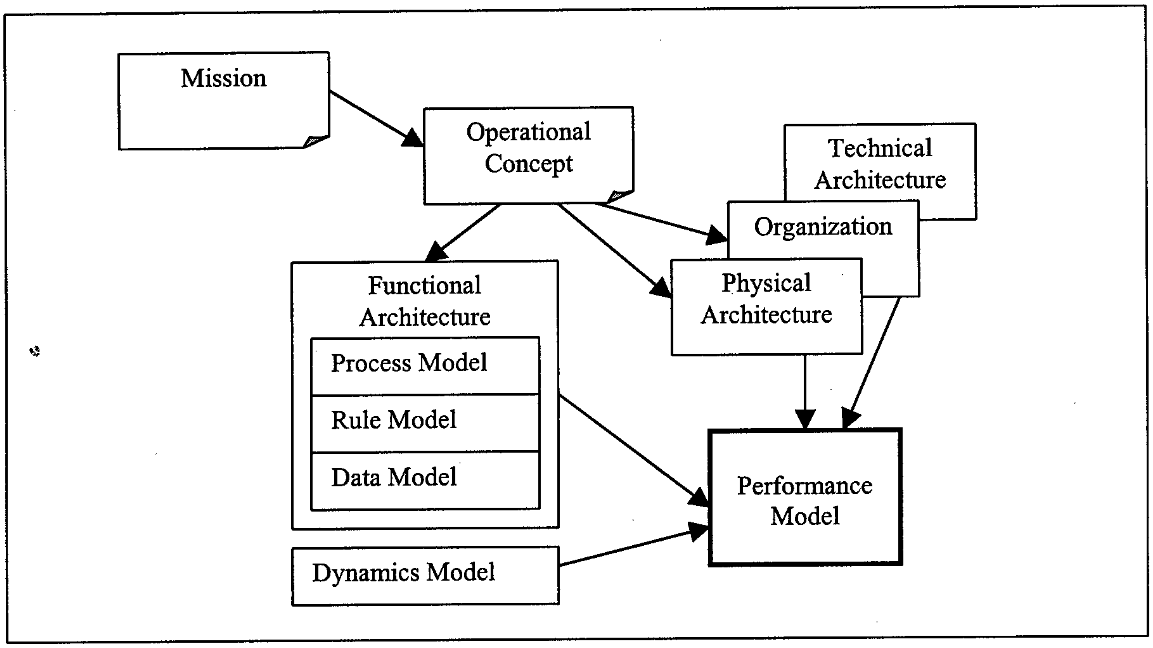

Figure 2. Architecture Views and Models

Performance Modeling in the System Engineering Process

The crux of the system engineering problem is to define a system that will perform efficiently as needed to deliver the required operational capabilities, across the range of likely operating conditions, within cost and schedule constraints. This typically involves an iterative design approach, illustrated in figure 3. Requirements statements are analyzed to define functional capabilities, operational and system structures, and performance measures directly linked to requirements. Once defined, the functions are allocated to system components and are assigned initial performance budgets. Alternative functional allocations, process flows, and resource levels are examined and analytic results documented. As the alternatives are evaluated the results flow into updated performance budgets, lower level requirements statements and the system specification. A key tool facilitating this process is the performance model used to address key questions and critical aspects of the system under development. Development and use of this performance model is the focus of this paper.

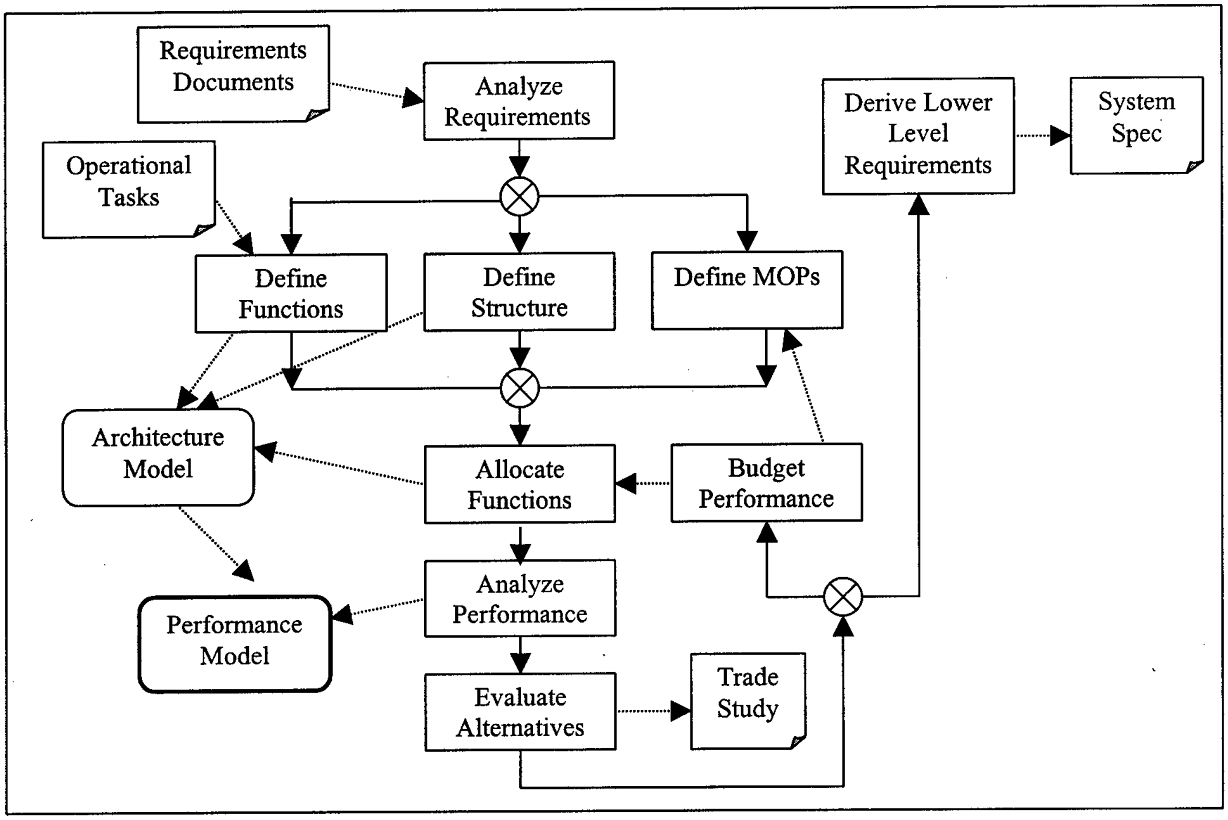

Figure 3. Simplified System Engineering Process

Performance Modeling

The utility of a dynamic discrete event simulation of the system under development is that it permits system developers and users to:

- Determine expected system behavior and performance before actually building the system
- Develop an understanding of why the system performs the way it does, and what might be done to make it perform better
- Locate impediments to system performance – such as buffers and queues that backup, time consuming processes; throughput degraded by slow processors or insufficient communications bandwidth
- Identify candidate processes for automation
- Investigate impact of system modifications
- Examine impact of rearranging the sequence in which processes are performed
- Estimate system resource requirements – human, machine, and material – and the impact of alternative resource levels
- Determine impact on system performance of varying degrees of system reliability
- Effectively explain the system concept to both the customers and the design engineers who will implement the planned capabilities
- Verify design engineers and customer's understanding of requirements

An often overlooked benefit, but not at all trivial, is the degree of insight into the system requirements gained through the process of model development. The model will represent the system as a series of primitives consisting of queues, time delays, probability distributions, equations, routers, processors, and so on. These are quite specific, showing what the system does, when, how, and why. When the engineers or operators walk through the model, every process, every decision point, every logical rule, and every assumption is examined. This in itself can be highly instructive, and will ultimately save both time and money.

MODEL DEVELOPMENT

Problem Definition

The first step is to define the issues of concern to the development team. This is a process of identifying the questions that need to be answered, the information needed to answer these questions, and the level of detail required. As the questions are developed, associated performance parameters – the factors that are important to system operation – will be identified for collection by the model. An important part of defining the question and determining the information the model must produce is specification of exactly where and when in the system process flow the performance data will be measured, and the kinds of plots and graphs to be generated from this data.

At the same time, the Measures of Performance (MOP) and Measures of Effectiveness (MOE) that will be used to assess system performance must be identified. The MOP is a measurement of the system itself without addressing the external environmental factors, while the MOE is a measurement of the response of the system with respect to its interaction with the external environment. Often the performance measures of interest will be such factors as time, rate, and accuracy. An example can be the revolutions per minute of an engine. These may be rolled into effectiveness measures such as throughput, capacity, responsiveness, and timeliness, to allow objective comparisons of system performance in terms that matter - that are keyed to requirements. The effectiveness measure for a car can be the acceleration or the handling of the car on the road where external conditions are being considered, for example.

It is important that the questions to be answered be framed as completely and accurately as possible, because they dictate the approach the modeler will use, the degree of detail required, and the outputs the analysis must generate. The questions will be refined as the effort progresses and insight into system operation is gained, but serious early thought will help ensure the modeling effort is correctly focused and scoped, and hence likely to be successful. The problem definition needs to include a description of the operational environment to be simulated, and the operational concepts governing system employment, because this is what drives the model, and ultimately, measures of system effectiveness.

Experiment Design

Having developed a listing of the questions of concern, a statement of analysis objectives and specific analysis requirements, it is appropriate now - prior to any model development - to design the experiments that will be conducted with the model. Experiment design should include the factors to be varied, the range of values to be tested, the environment or scenarios to be used, and the number of simulation runs to be made. This should be documented in an experiment design document. Although the experiment plan will almost certainly be refined, it is important that it be drafted now, to ensure that the model will be designed such that it will be capable of producing the necessary information.

It is very useful at this beginning point to develop, with participation of all the stakeholders, an informal statement of expectations, or "success criteria," to help

ensure the modelers understand the objectives of their efforts. Model scope, boundaries, limitations, and level of detail should be mutually understood and agreed before going further.

System Representation Requirements

The system representation requirements amount to a description of what is to be modeled. This typically takes the form of functional flow diagrams and descriptions of the processes being performed, together with inputs, outputs, controls, decision logic, interfaces, and the resources employed to perform the functions. These need not be fully fleshed out, and probably will not be, and in any event are likely to change as a result of behavior modeling. What is essential is an understanding of system components, interactions, and the factors that may vary as the system operates, or cause the system to respond in different ways.

However it is accomplished, the following must be defined to the level of detail needed to address the issues of concern:

- The operational entities and system nodes to be represented, together with their relationships to one another, and the connectivity among them;
- The information that is exchanged among architecture nodes, and is important to the elements of system performance being examined, must be defined. This should include (as appropriate to the question), data type and size, frequency with which it is exchanged, data path employed, intended distribution, and the attributes which give meaning to the data;
- Operational concepts governing or describing system operation;
- System functional flow: the processes performed, their sequence, their inputs, outputs, controls, the resources to accomplish each process, and the logic governing each decision made within a process;
- Subsystem or system element performance parameters and values: factors such as bandwidth, message size, communication protocols, data rate, delay time to accomplish each function, buffer capacity, and any other factors important to system performance.

A model should not be any more complex than is necessary to answer the question for which it is built. A key to effective and efficient use of modeling resources in support of system engineering and design activities is to model only the parts of the system that are important to answering the questions of concern, and to model them only to the depth necessary. More detail is not necessarily better, but it does make the model harder to use and understand. The modeling activity should focus on key system functions, the identification of which will vary according to the questions being addressed. In C4ISR systems, "key functions" are often those that require significant time or other resources, lie in the critical path, require completion of a human decision process, or produce an output upon which a key function is dependent for execution. Model simplification can be achieved by carefully assessing the question, and omitting less important factors, aggregating several processes, or characterizing processes, rather than explicitly modeling them. An incremental approach is often effective, in which the initial model is a fairly simple approximation of the system. This model will be iteratively refined and enhanced as design concepts mature and greater understanding and insight, into both the system and the question, is developed.

The system representation requirements constitute the "specification" to which the model will be built and verified.

Model Architecture Review

When the model is completed to the point that it incorporates system nodes, elements, processes and information flow, it should be jointly reviewed in a working level session. This review needs to include the system architect, the system engineers, their counterparts in the customer organization, and the model developers. It should be a detailed, block by block walk through of each process and logical operation. The point of this review is to achieve consensus on the model's characterization of system behavior and connectivity.

Many modeling tools available on the open market provide basic building blocks that can be assembled to represent the functional process being performed. These building blocks include graphic objects representing queues, delays, probability distributions, and so on, thus simplifying the review process. As an example, figure 4 shows a partial model (using the COTS tool Extend) of a radio in which an incoming report is placed in a queue, held for a processing delay (delay time read from the input spreadsheet), then routed through a switch which is set based on a threshold value. Subsequently the report is screened and either routed on, or dropped, based on a probability distribution. The graphic display allows someone who is not a modeler but does understand how the system should work to understand and verify the modeler's representation of the functional processes.

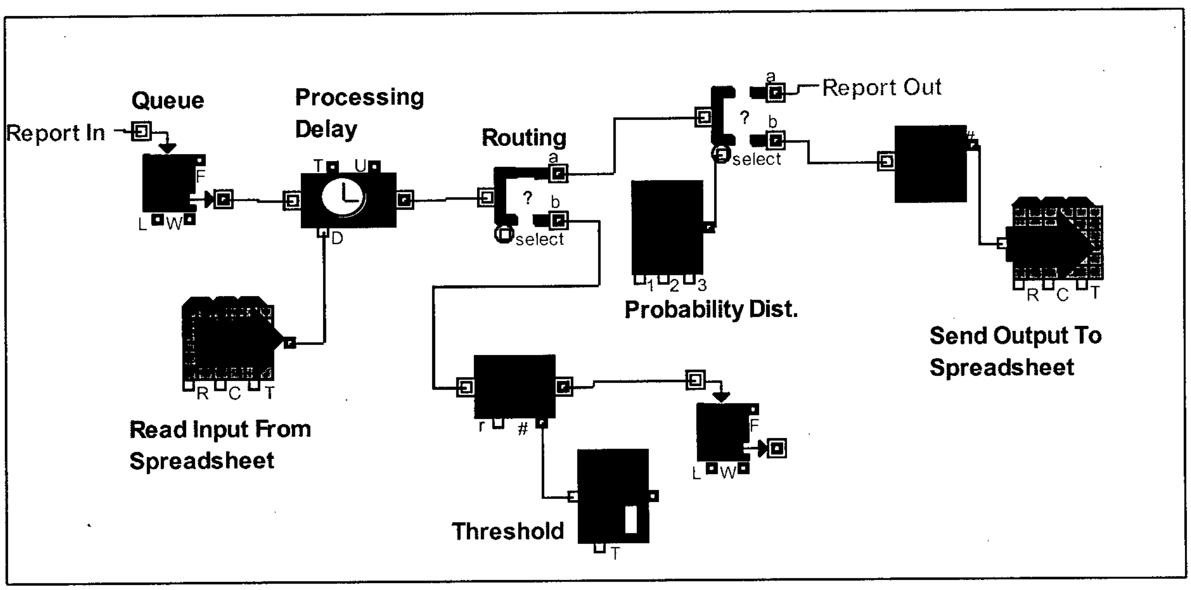

Figure 4. Sample Model

Throughout the review, it is important to keep a focus on the purpose of the modeling effort. The model is supposed to answer certain questions, not all questions. Consequently, parts of the model will be developed to greater detail than other parts. The idea is to be sure that the right parts are developed to sufficient detail to provide the needed information. For example, in a model developed to examine the timeliness and accuracy of the tactical data exchange among a group of ships and aircraft, the communications network would be developed in detail, but the ships' sea

keeping abilities would probably be addressed in only a cursory way, if at all. This is important to keep in mind first when the representation requirements are developed, and later when the model is accredited for use – it could be properly accredited for certain purposes, but not others.

In a complex development there should be multiple reviews, perhaps as incremental portions of the model are completed. The idea is that the modeling effort should not be allowed to proceed completely independently; it is too easy to make a wrong turn and waste time and effort building something that is not quite what the customer wanted (even if it is what was requested).

Model Parameters Development

During model development, many of the input parameters, such as process time delays, will not be immediately available. Model construction can proceed in their absence, but eventually the model needs to be populated with appropriate parameter values. These can be developed by accessing results of engineering tests, specification values, prototyping activities, engineering estimates, or budget values as appropriate.

During the experimentation phase the system analysts will want to change subsystem performance values to observe their effect on overall system performance. An effective approach is to link the model to a spreadsheet, and read performance values into the model when the simulation is initialized. This has several benefits. The performance values used in the simulation are all consolidated in one place, are readily visible, can be documented with data source references, and can be changed without going into the model to find all the places the values might be used. This simplifies the process of setting up subsequent experiments, when individual process performance values are often varied to observe their impact on system performance. The same spreadsheet also provides a logical means of collecting output data, as simulation runs are completed, and a means of capturing the input conditions tied to the results of each experiment.

Model Instrumentation

Values generated by the model must be captured to assess system performance and effectiveness. Usually these values are related to timing, accuracy, process and resource utilization, queue length and wait time, item arrivals and departures at various points in the model, and changes in attribute values. The values are generated within the model, as each simulation item passes through the various modeled processes. Model instrumentation means the insertion of data collection blocks at each point of interest. A variety of forms of instrumentation are available, such as plotters, histograms, blocks that capture minimum and maximum values, or generate statistical data concerning the items that pass through the block. The model can be configured to write results data into external files, including spreadsheet and text files, for offline analysis.

Instrumenting the model can be time consuming, and to be effective the customer must identify his data needs before the modeler sets to work. Few things are more frustrating than to amass a pile of data only to learn that the user wanted to know when the first round leaves the muzzle, not when the last round impacts the target. The means by which instrumentation is accomplished varies for different modeling tools. In Extend, instrumentation means insertion of blocks that collect, compute and display measures of performance. Generally, model instrumentation is analogous to insertion of probes that allow users to examine certain state parameters either as output values or to provide insight for further analysis.

Model Testing

Testing is performed incrementally, checking each process block to ensure that the parameters, information flow, and inter-connectivity are free of implementation errors. Item attribute and value information is extracted within each block to verify that proper values are being generated. Once all of the hierarchic process blocks have been successfully tested, the entire model is executed and observed data checked for reasonableness. Input parameters are varied, and tests repeated to ensure that no data - dependent model construction errors have been made.

Documentation

As usual, clear, concise, and complete documentation is essential. There are at least two kinds of documentation involved. The first has been mentioned already – documenting the values of the parameters used within the model, which helps in the verification, validation, and accreditation process and is necessary to interpret experiment results and maintain the model. The second form of documentation has a somewhat different purpose, which is to make the model easy to understand and use. The model should be annotated internally, with text that provides a brief description of the process being modeled. For example, one might insert a comment within the appropriate portion of the model such as "Tomahawk mission data is assembled into Transmission Units here." In addition, each of the primitive blocks used to construct the model should be commented with a note concerning their purpose in the model, such as "This queue holds items representing Mission Data Updates until the next process is ready to accept them for transmission." If the model is properly commented, it becomes easy to understand both how the modeled system is supposed to work, and what each of those little model blocks are doing. When this is done, it is a lot easier to explain the model to the customer/ user, and to verify the model's representation of system processes. It also makes it reasonable to expect that the model could be transferred from the model developer to a member of an analysis or engineering team (who are not modelers), who can then use it to conduct experiments as desired.

MODEL VERIFICATION, ACCREDITION AND USE

Verification

Model verification is "the process of determining that a model implementation accurately represents the developer's conceptual description and specifications." (DMSO 1996) Verification is undertaken to develop confidence that the model correctly represents system behavior, and can accurately predict the performance of the planned system. There are two aspects to this; first assurance that the model behaves in the same manner as the planned system, and second that the performance parameters used to populate the model are appropriate for the purpose at hand.

The first step in model verification is to ensure that the model is built and represents the planned system faithfully, that the operational and functional processes represented have been accurately captured. This typically is accomplished through interaction with subject matter experts – the people specifying or developing the system, and current/planned operational users of the systems – and through examination of existing design and operational concept documentation. This activity involves verifying that each process is correctly characterized, that the control events, data flows into and out of the processes, the logic exercised within each process, and the resources performing the processes are modeled in a way that captures their essential characteristics. Checks should be made to ensure that the behavior symptoms generated by the model are consistent with real system behavior under similar circumstances. For example, if a legacy system is being modeled and is known to

bog down when loading reaches a certain point, then the model should as well. There will be many situations in which there are no real systems that are analogous to the modeled system. In such cases, one will have to verify the behavior for each component or part of the model until all of its components have been verified.

Assumed performance values contained within the model must be certified to be valid (where appropriate) or to satisfy the analysis requirement. The distinction here is that performance values can only be demonstrated to be valid after the system is actually built and tested. However, one can and must verify that the input performance data for a planned system is reasonable for the analysis being performed. Values used normally range from measured prototype or subsystem test data collected under controlled conditions, to performance specification values, to estimates offered by the project engineers or experienced operators. These often are budgetary values, or ranges of values to be evaluated. In the case of interactive systems, an operator's opinion is often the best and sometimes the only data available. As an example, consider the requirement to model a signals intelligence process which includes receipt of a communications signal, its translation into English, assessment of the importance of the message, and generation of a report to the tactical commander. Time to accomplish this process depends on many things, not least of which is the language being spoken, the skill of the linguist, and the intensity of the operations. The only way to determine this for model development was to interview the actual operators. When asked, the operator said that it took her at least x minutes, but usually not more than y minutes, which was then modeled with an appropriate probability distribution.

As the system evolves and designs are refined, additional factual information becomes available and will be used to refine the model. If model outputs can be corroborated with real or prototype component test data for some situations or configurations, then the likelihood that predicted performance under other conditions would be accurate is enhanced. A particularly effective form of model validation and tuning is one in which modeling activities are undertaken in conjunction with system prototyping, so that the model is continually refined as system design uncertainties are reduced.

Accreditation

Finally, an official determination that the model is (or is not) acceptable for the specified use must be made. This is accreditation. The accreditation report documents the conclusion that the model plus the input data equals a valid representation of the planned system for the purpose at hand. If the model is later proposed for reuse in a different context, to address different questions, it will need to be re-examined and re-accredited.

Using the Model

The experiments conducted with the model are intended to answer a set of questions. These questions might be phrased as "How well does the system work? When does it break? How can it be made to work better? What happens if . . ." These questions concern the ability of the system design to meet warfighting objectives, which the analysts and system engineers have decomposed into measurable requirements. Performance measures, such as accuracy, timeliness, throughput, reliability, are assessed against performance requirements to yield measures of effectiveness. The model is used to collect performance metrics that can be related back to warfighting requirements, system effectiveness, and associated design parameters.

A typical series of experiments will first establish the effectiveness of the baseline system in meeting warfighting objectives across a range of environments or operational situations. Then, individual

performance metrics are examined to identify design factors which limit system effectiveness. Alternative solutions are offered, and the experiments repeated with the modified system design or system performance parameters. Generally, the effort is to identify expected system performance and robustness, the elements of the system design that are sensitive to changes in the external environment, and the design factors which most significantly impact system performance.

This is the point at which the experiment plan - which was developed before the model was constructed – is executed and the results documented.

AN EXAMPLE

Several years ago significant concerns were expressed regarding the ability of deployed tactical radios and tactical data processors (TDPs) to deal with increased data flows associated with the introduction of new tactical broadcasts providing situation awareness information. The new broadcasts were bursty in nature; providing relatively high volumes of information when active but operating on an intermittent schedule. The technical issue concerned data rate differentials – when available, the broadcasts might provide information faster than it could be assimilated by the radios and data processors. The questions then, were whether tactical data would be lost in the receiving and processing equipment, and/or delayed to the point that its tactical value was degraded. There was much discussion of the issue, and many opinions were offered, but factual answers were lacking. When the systems involved were modeled using a discrete event simulation tool, it was predicted that both problems would occur – data losses would approach 50% during peak periods, and time delays of nearly 45 minutes would be experienced. Input data would queue up in the radio pending processing, the buffer would soon fill and new reports would overwrite earlier ones. The TDP was unable to read the over-the-air message format, and converting the native format to one readable by the TDP significantly expanded the number of bytes of data requiring transfer. To reduce the problem in the TDP, the interface between the radio and the TDP was constrained to a low data rate. This meant that less data was lost in the TDP, but it simply backed up and was lost in the radio instead of the TDP. When the data did make it to the TDP, the slow processor and limited data storage allocation caused losses at the input buffer, and later in the process flow, limited memory caused tracks to be overwritten far too quickly. This was all made very obvious in the model.

As a result of the modeling effort, one TDP developer made a number of design changes to his system:

- Increased processor clock speed
- Increased system RAM
- Doubled memory space allocated to report and track data bases
- Increased interface speed to the highest rate supportable by the radio
- Defaulted the input message format to the most efficient available

These hardware and software changes eliminated the data loss problem before it was ever experienced by a deployed system produced by this developer. However, analogous changes were not implemented by other programs/ system developers. Nearly three years later, and as a result of concerns expressed by the operating forces, live tests using operational hardware were conducted. These tests demonstrated that the problems predicted by the simulation in 1997 were being experienced in other deployed systems in 1999.

SUMMARY

Focused management attention to the following key areas will help ensure a successful project using modeling and simulation:

- Carefully define the questions of concern and the information the model must produce to answer the questions. Certainly the questions will evolve as results are obtained, but they should become more refined and specific, rather than changing focus entirely.
- Determine the degree of model fidelity required, and base it on a definition of the information the model must produce. Simpler is better.
- Select performance measures that can be tied directly to requirements and design parameters.
- Carefully chose the performance metrics, and specify where in the process flow they should be collected.
- Structure the effort as an incremental development. Make enhancements as system sensitivities become known and the issues clarify.
- Conduct regular model walk-throughs during initial development and when significant changes are made.
- Develop a matrix of the experiments to be conducted. Include in this matrix a specific and detailed list of each parameter to vary, and the range of values to be tested.

CONCLUSION

Performance modeling is an essential element in development and understanding of complex systems. Effective use of modeling techniques and tools can provide, in advance of actually building and deploying the system, engineering insight available in no other way. Some cautions are in order, however. Developing and experimenting with a behavior model can consume significant time and resources. Some careful planning at the beginning of the effort, with refinement as the work proceeds, will ensure a more effective, efficient and useful project and product.

REFERENCES

Coyle, Dr. Philip and Dr. Patricia Sanders, Simulation, Test, and Evaluation Process (STEP) Guidelines, December 1997.

C4ISR Architecture Framework Version 2.0, C4ISR Architecture Working Group, 18 December 1997

Verification Validation and Accreditation Recommended Practices Guide, Defense Modeling and Simulation Office, US Department of Defense, November 1996, p1-5.

Steve Froggett's background includes 20 years service as a Naval officer in surface combatants, afloat staffs, OPNAV, and finally as Vertical Launch TOMAHAWK Program Manager for the Cruise Missile Program. After retiring from the Navy in 1988, he was Director Washington Operations for Tiburon Systems, Texas Instruments Advanced C3I Systems, and Raytheon Systems Company Advanced C3I Systems, in succession. (The succession represents purchase of one company by the next). He is currently employed by ComGlobal Systems, Inc. (ComGlobal has not purchased Raytheon).

Simon Liu is a senior software and systems engineer with extensive experience in software development and system modeling of command and control and weapon systems. He has worked for Lockheed Missiles and Space Company (now Lockheed Martin) and Tiburon Systems, which through acquisitions became Texas Instruments and Raytheon Systems Company. He is currently employed by ComGobal Systems Inc.